



# A Rodin plugin for quantitative timed models

Joris Rehm

## ► To cite this version:

Joris Rehm. A Rodin plugin for quantitative timed models. Rodin User and Developer Workshop, Jul 2009, Southampton, United Kingdom. inria-00431246

**HAL Id: inria-00431246**

**<https://inria.hal.science/inria-00431246>**

Submitted on 11 Nov 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Rodin plugin for quantitative timed models\*

Joris Rehm

joris.rehm@loria.fr - LORIA - Nancy Université

We propose to develop a Rodin<sup>1</sup> plug-in that experiments a systematic use of a refinement pattern. The goal of this pattern is to help in modeling of timed system in Event-B. By timed system we mean system with quantitative temporal constraints and properties.

The user (of the plug-in) will see and modify an Event-B machine augmented with a new operator  $S$ . This unary operator over an event name  $e$  gives the delay elapsed since the latest triggering of the event  $e$ . By using this operator in the invariant, the user will be able to write and prove temporal properties over the events. By using this operator in the guard of an event  $f$ , the user will be able to specify temporal constraints over the duration  $S(e)$  (which becomes here the delay between the event  $e$  and  $f$ ). For the possible constraints, we plan to consider lower time bounds ( $l \leq S(e)$  in guard of  $f$ ) and upper time bound ( $S(e) \leq u$  in guard of  $f$ ). The upper time bound is a bound within the event  $f$  must obligatory occurs, it is not just a possibility. In this text, the usages of the operator  $S$  is called the annotations, it is an extension of the syntaxe of the B models. The annotations can only appear in invariant or guards.

Our pattern is an Event-B model that encodes the behaviour of the operator  $S$  (we call this model the pattern model). The annotations given by the user define how to refine the pattern model and how to obtain the behaviour of  $S$  needed for a particular augmented model. The goal of our plug-in is to generate a normal Event-B model that is the studied (augmented) model where the annotations are replaced with the superposition of the refined pattern model.

To explain briefly what is the idea of the pattern, we show below an augmented model (on the left) and the generated result (on the right). This small model define a light that can be on ( $lo = TRUE$ ) or off ( $lo = FALSE$ ); the light can be switch on by a button (the event  $on$ ) and goes automatically off (event  $off$ ) after a delay between  $c - d$  and  $c + d$ . You can see the temporal annotations in the elements named *lb\_off* (Lower Bounds) and *up\_off* (Upper Bounds). The  $S$  operator can also appear in the invariant, for example we can write  $c + d < S(on) \Rightarrow lo = FALSE$ .

EVENTS	EVENTS
$on \triangleq$	$on \triangleq$
<b>Begin</b>	<b>Begin</b>
act1: $lo := TRUE$	act1: $lo := TRUE$
<b>End</b>	act2: $s_{on} := 0$
	<b>End</b>

\*This work was supported by grant No. ANR-06-SETI-015-03 awarded by the Agence Nationale de la Recherche.

<sup>1</sup><http://www.event-b.org>

